49265

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Application of : **HAYARDENY et al.**

Serial No. : 10/673,733 : Group Art Unit: 2161

Filed : September 29, 2003 : Examiner: Paul Kim

For : STORAGE DISASTER RECOVERY USING A PREDICTED SUPERSET OF UNHARDENED PRIMARY DATA

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

DECLARATION UNDER 37 CFR 1.131

Sir:

We, the undersigned, Amiram Hayardeny, Martin Tross and Aviad Zlotnick, hereby declare as follows:

1) We are the Applicants in the patent application identified above, and are the inventors of the subject matter described and claimed in claims 1-60 therein.

2) Prior to August 29, 2003, we conceived our invention, as described and claimed in the subject application, in Israel, a WTO country. Prior conception of the invention is evidenced by a draft of the patent application sent to us on August 25, 2000, by Dr. Daniel Kligler, of Sanford T. Colb & Co., who was retained by

IBM as outside counsel for the purpose of preparing the present patent application. A copy of the draft, with Dr. Kligler's cover letter, is attached hereto as Exhibit A. We note that the independent claims in this draft are nearly identical to those in the application as filed.

3) Shortly after receiving the draft, we reviewed it and gave Dr. Kligler our comments. Dr. Kligler then sent us a revised draft on September 3, 2003, which we approved immediately. Dr. Kligler sent the final draft for filing to Suzanne Erez in the IP Department of the IBM Haifa Research Laboratory on September 4, 2003. A copy of Dr. Kligler's cover letter is attached hereto as Exhibit B.

4) Ms. Erez immediately sent the application to the IBM Thomas Watson Research Center for filing by an IBM attorney in the USPTO. Filing was deferred, however, because the heading of the formal drawings did not comply with IBM standard practice. Ms. Erez conveyed a request to Dr. Kligler to correct the drawings on September 14, 2003. Dr. Kligler's draftsman, Ronen Harel, sent the corrected drawings to Ms. Erez on September 15, 2003. A copy of Mr. Harel's cover letter is attached hereto as Exhibit C.

5) Ms. Erez passed the corrected drawings on to the IBM Thomas Watson Research Center, and the application was then filed on September 29, 2003.

We hereby declare that all statements made herein of

our knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

Amiram Hayardeny — December 14, 2006

Amiram Hayardeny, Citizen of Israel — Date
Sun China Engineering & Research Institute
7/F Chuangxin Plaza
Tsinghua Science Park
Beijing 100084

_____ _____

Martin Tross, Citizen of Israel — Date
5 Geulei Teiman Street,
Haifa 34991 Israel

_____ _____

Aviad Zlotnick, Citizen of Israel — Date
Mizpe Netofa
D.N. Galil Takhton, Israel

Exhibit A

From: Daniel Kligler
To: Aviad <aviad@il.ibm.com>
Date: 8/25/03 10:43AM
Subject: IL9-2003-0031, our ref. 49265 - IBM CONFIDENTIAL

Dear Aviad,

Attached please find a first draft of this application (text and figures) - the next in the series of your storage inventions.

Please review this draft, and let us have your corrections and comments at your earliest opportunity. Note a number of questions to you that I have marked in boldface in the text.

I am still waiting for your feedback on the revised draft of IL9-2002-0033 (our 49267) and on the first draft of -0032 (our 49266). If I am far off the mark in either of these cases, please let me know. Otherwise, I will go ahead now to prepare the last application in this series (-0028, our 49264).

Regards,
Danny

CC: Suzanne Erez

STORAGE DISASTER RECOVERY USING A PREDICTED SUPERSET OF UNHARDENED PRIMARY DATA

CROSS-REFERENCE TO RELATED APPLICATION

This application is related to a U.S. patent application filed on even date, entitled "Asynchronous Data Mirroring with Look-Ahead Synchronization Record" (IBM docket number IL9-2003-0032), whose disclosure is incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates generally to data storage systems, and specifically to data mirroring for failure protection in storage systems.

BACKGROUND OF THE INVENTION

Data backup is a standard part of all large-scale computer data storage systems (and most small systems, as well). Data written to a primary storage medium, such as a volume on a local storage subsystem, are copied, or "mirrored," to a backup medium, typically another volume on a remote storage subsystem. The backup volume can then be used for recovery in case a disaster causes the data on the primary medium to be lost. Methods of remote data mirroring are surveyed by Ji et al., in an article entitled "Seneca: Remote Mirroring Done Write," *Proceedings of USENIX Technical Conference* (San Antonio, Texas, June, 2003), pages 253-268, which is incorporated herein by reference. The authors note that design choices for remote mirroring must attempt to satisfy the competing goals of keeping copies as closely synchronized as possible, while delaying foreground writes by host

processors to the local storage subsystem as little as possible.

Large-scale storage systems, such as the IBM Enterprise Storage Server (ESS) (IBM Corporation, Armonk, New York), typically offer a number of different copy service functions that can be used for remote mirroring. Among these functions is peer-to-peer remote copy (PPRC), in which a mirror copy of a source volume on a primary storage subsystem is created on a secondary storage subsystem. When an application on a host processor writes to a PPRC volume on the primary subsystem, the corresponding data updates are entered into cache memory and non-volatile storage at the primary subsystem. The control unit (CU) of the primary subsystem then sends the updates over a communication link to the secondary subsystem. When the CU of the secondary subsystem has placed the data in its own cache and non-volatile storage, it acknowledges receipt of the data. The primary subsystem then signals the application that the write operation is complete.

PPRC provides host applications with essentially complete security against single-point failures, since all data are written synchronously to non-volatile media in both the primary and secondary storage subsystems. On the other hand, the need to save all data in non-volatile storage on both subsystems before the host write operation is considered complete can introduce substantial latency into host write operations. In some large-scale storage systems, such as the above-mentioned IBM ESS, this latency is reduced by initially writing data both to cache and to high-speed, non-volatile media, such as non-volatile random access memory (RAM), in both

the primary and secondary subsystems. The data are subsequently copied to disk asynchronously (an operation that is also referred to as "hardening" the data) and removed from the non-volatile memory. The large amount of non-volatile memory that must be used for this purpose is very costly.

SUMMARY OF THE INVENTION

The present invention provides methods for data mirroring that can be used to create storage systems that are immune to single-point failures, have low-latency write response, and permit rapid recovery after failure, without requiring special non-volatile memory or other costly components.

In embodiments of the present invention, a storage system comprises primary and secondary storage subsystems, which are configured to back up data from the primary to the secondary subsystem over a communication link in an asynchronous mirroring process, whereby the data are ultimately stored in non-volatile media on both the primary and secondary subsystems. The mirroring process is controlled using a metadata record, held on the secondary subsystem, which identifies storage locations that may be "out of sync" (i.e., may contain different data) on the primary and secondary subsystems. The metadata record is maintained in such a way that the locations identified in this record constitute a predictive superset of the locations that are actually out of sync. Typically, the primary subsystem is aware of the contents of this record, as well.

When a host writes data to a specified location on the primary subsystem, the primary subsystem places the data in its cache memory. If the specified location is already included in the metadata record (i.e., the record indicates that the location may be out of sync), the primary subsystem signals the host immediately to indicate that the write operation has been completed. Otherwise, if the specified location is not listed in the metadata record, the primary subsystem sends a message to

the secondary subsystem, which causes the secondary subsystem to update the record synchronously. This message may include the actual data to be copied from the primary to the secondary subsystem. The primary subsystem in this case signals the host that the write operation has been completed when the primary subsystem receives an acknowledgment of this message from the secondary subsystem. In either case, the primary system acknowledges the write to the host without waiting for the data to be actually written to the disk (or other non-volatile media) on either the primary or secondary subsystem. The latency of write operations is thus held to a minimum. Because only a fraction of the host write operations (generally a small fraction) cause the primary system to send synchronous messages to the secondary subsystem, the added burden of communication traffic on the communication link between the subsystems is small.

Upon recovery from a failure on the primary subsystem, the secondary subsystem uses the metadata record to determine the locations from which it should copy data back to the primary subsystem in order to ensure that the two subsystems are fully synchronized. Because the metadata record is predictive, it may include some locations that are actually in sync, and will therefore be copied back unnecessarily. By judicious maintenance of the metadata record, as described hereinbelow, the amount of unnecessary copying can be limited, so that the primary subsystem recovers rapidly. The size of the predicted superset may be controlled so as to achieve the desired balance between write latency (which becomes shorter as the predictive superset is

enlarged) and recovery time (which becomes shorter as the superset is reduced).

{Claim summary will be added here in the final version.}

The present invention will be more fully understood from the following detailed description of the preferred embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram that schematically illustrates a data storage system, in accordance with an embodiment of the present invention;

Fig. 2 is a schematic representation of data structures used in tracking data storage, in accordance with an embodiment of the present invention;

Fig. 3 is a flow chart that schematically illustrates a method for tracking data storage, in accordance with an embodiment of the present invention; and

Figs. 4 and 5 are flow charts that schematically illustrate methods for maintaining a predictive metadata record, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Fig. 1 is a block diagram that schematically illustrates a data storage system 20, in accordance with an embodiment of the present invention. System 20 comprises storage subsystems 22 and 24, which are labeled "storage node A" and storage node B" for convenience. In the description that follows, it is assumed that node A is configured as the primary storage subsystem, while node B is configured as the secondary storage subsystem for purposes of data mirroring. Thus, to write and read data to and from system 20, a host computer 26 (referred to alternatively simply as a "host") communicates over a communication link 28 with subsystem 22. Typically, link 28 is part of a computer network, such as a storage area network (SAN). Alternatively, host 26 may communicate with subsystem 22 over substantially any suitable type of serial or parallel communication link. Although for the sake of simplicity, only a single host is shown in Fig. 1, system 20 may serve multiple hosts. Typically, in normal operation, hosts may write data only to primary storage subsystem 22, but may read data from either subsystem 22 or 24.

Subsystems 22 and 24 may comprise substantially any suitable type of storage device known in the art, such as a storage server, SAN disk device or network-attached storage (NAS) device. Subsystems 22 and 24 may even comprise computer workstations, which are configured and programmed to carry out the storage functions described herein. Subsystems 22 and 24 may be collocated in a single facility or, for enhanced data security, they may be located at mutually-remote sites. Although system 20 is shown in Fig. 1 as comprising only a single primary

storage subsystem and a single secondary storage subsystem, the principles of the present invention may be applied in a straightforward manner to systems having greater numbers of primary and/or secondary storage subsystems. For example, the methods described hereinbelow may be extended to a system in which data written to a primary storage subsystem are mirrored on two different secondary storage subsystems in order to protect against simultaneous failures at two different points.

Each of subsystems 22 and 24 comprises a control unit (CU) 30, typically comprising one or more microprocessors, with a cache 32 and non-volatile storage media 34. Typically, cache 32 comprises volatile random-access memory (RAM), while storage media 34 comprise a magnetic disk or disk array. Alternatively, other types of volatile and non-volatile media, as are known in the art, may be used to carry out the cache and storage functions of subsystems 22 and 24. The term "non-volatile storage media," as used in the context of the present patent application and in the claims, should therefore be understood to comprise collectively any and all of the non-volatile media that are available in a given storage subsystem, while "cache" or "volatile memory" comprises any and all of the volatile media. Control units 30 typically carry out the operations described herein under the control of software, which may be downloaded to subsystems 22 and 24 in electronic form, over a network, for example, or may be provided, alternatively or additionally, on tangible media, such as CD-ROM.

Subsystems 22 and 24 communicate between themselves over a high-speed communication link 36, which may be part of a SAN or other network, or may alternatively be a dedicated line between the two subsystems. Subsystem 24 may also be coupled to communicate with host 26, as well as with other hosts (not shown), over a communication link 38, similar to link 28. Link 38 enables subsystem 24 to serve as the primary storage subsystem in the event of a failure in subsystem 22. **{During failover, how does the secondary subsystem handle the tracks that are marked in its MOOS? The host may have written to some of these tracks on the primary, and received a write acknowledgment from the primary, without the data having been copied yet to the secondary. When the host tries to read from these tracks on the secondary, it may get unexpected results.}** It will be thus be observed that the capabilities of the primary and secondary storage subsystems are substantially identical, and the functional designations "primary" and "secondary" are arbitrary and interchangeable. Optionally, subsystem 22 may serve as the primary subsystem for some hosts, while subsystem 24 serves as the primary subsystem for others, at the same time as it serves as the secondary subsystem for backup of subsystem 22. In this latter case, subsystem 22 may serve as the secondary subsystem for backup of subsystem 24.

Since subsystem 24 is intended to provide reliable backup even in case of a disaster at the site of subsystem 22, subsystem 24 may be held in particularly secure conditions at a remote site. The remote site may be maintained by an outside service provider, who

provides secure storage on a fee-per-service basis to the owner of subsystem 22 and to other storage users. Alternatively, control unit 30 and cache 32 of subsystem 24 may be collocated with subsystem 22, while storage media 34 of subsystem 24 are remotely located, as described in a U.S. patent application entitled, "Low-Cost Remote Data Mirroring" (IBM docket number IL9-2003-0033), filed _____, 2003, whose disclosure is incorporated herein by reference.

In the embodiments described below, it is assumed that system 20 is configured for asynchronous data mirroring. In other words, upon receiving data from host 26 to be written to subsystem 22, control unit 30 writes the data to cache 32, and then signals the host to acknowledge the write operation without necessarily waiting for the data to be copied to secondary subsystem 24. Control unit 30 stores the data from cache 32 to its local storage media 34 and transmits the data over link 36 to subsystem 24 for mirror (backup) storage by means of background processes. After receiving the data, or possibly after storing the data at the appropriate locations on its own storage media 34, control unit 30 of subsystem 24 sends an acknowledgment back to subsystem 22. The data storage and mirroring operations on subsystem 24 is thus carried out asynchronously and independently of the completion of the write operation between host 26 and subsystem 22.

Fig. 2 is a schematic representation of data structures that are maintained on subsystem 22 for tracking data storage in system 20, in accordance with an embodiment of the present invention. Bitmaps 40, 42 and 44 are metadata records, which are typically held in

volatile memory, such as cache 32, on subsystem 22. The bitmaps are used in recording the locations at which the data on storage media 34 in subsystems 22 and 24 are or may be out of sync. Each bit represents a different location. Typically, when storage media 34 comprise disks, each bit in the bitmaps corresponds to a disk track, but the bits (and the corresponding locations) may alternatively correspond to different sorts of data elements, of finer or coarser granularity. Furthermore, although the bitmaps described here are a convenient means for maintaining metadata records, other types of data structures may similarly be used for the purposes of the present invention, as will be apparent to those skilled in the art.

The specific purposes of the bitmaps shown in Fig. 2 are as follows:

- Bitmap 40 indicates "dirty" locations in cache 32 on subsystem 22. The marked locations in bitmap 40 correspond to tracks on storage media 34 in subsystem 22 to which host 26 has written data, which are currently being held in cache 32 and have not yet been hardened to the storage media. Control unit 30 of subsystem 22 clears the bits in bitmap 40 when it hardens the corresponding data, indicating that the cache is now clean with respect to these locations.
- Bitmap 42 indicates locations at which the data held by subsystem 22 (in cache 32 or storage media 34) are out of sync with the corresponding locations in the cache or storage media in subsystem 24. When control unit 30 of subsystem 22 signals host 26 to acknowledge a write operation to a specified track before having sent the

data over link 36 to subsystem 24, the control unit sets the bit corresponding to this track in bitmap 42. The control unit then sends the data to subsystem 24, and clears the bit when it receives an acknowledgment from subsystem 24 that the data have been received there. Bitmap 42 is therefore referred to as the "out-of-sync" (OOS) bitmap. Clearly, if subsystem 22 fails and then subsequently recovers, any locations marked by bits that were set in bitmap 42 at the time of failure must be copied back from subsystem 24 to subsystem 22 in order to synchronize storage media 34 on the two subsystems.

- Bitmap 44 contains a predictive superset of the bits set in bitmap 42, indicating both locations that are known to be out of sync with subsystem 24 and locations that are not currently out of sync, but to which host 26 is expected to write in the near future. Therefore, bitmap 44 is referred to as the "maybe-out-of-sync" (MOOS) bitmap. All bits that are set in bitmap 44 are also set by control unit 30 of secondary subsystem 24 in a similar MOOS bitmap held on the secondary subsystem. As described below, however, certain bits may be cleared in bitmap 44 on subsystem 22 before they are also cleared on subsystem 24. Thus, the bits that are set in the MOOS bitmap on the secondary subsystem are a superset of those set in bitmap 44 (wherein the superset may be identical to the subset).

Control unit 30 on subsystem 22 may use a generation table 60 to keep track of the bits that are set and cleared in bitmap 44. The use of the generation table is also described in detail hereinbelow.

Upon recovery of subsystem 22 from a failure, control unit 30 in subsystem 24 reads its own copy of the MOOS bitmap in order to determine the tracks that are to be copied back to subsystem 22 from subsystem 24. Subsystem 24 then transmits back the contents of these tracks, along with any other tracks that changed on subsystem 24 while subsystem 22 was out of service (if, for example, subsystem 24 was used as the primary storage subsystem during the failure and received additional write operations from host 26). During normal operation, the bits to be set in bitmap 44 are selected using a predetermined, predictive method, as described hereinbelow. The same method is used to set the bits in the MOOS bitmap on subsystem 24, thus ensuring that the bits set in the MOOS bitmap on subsystem 24 will always be a well-defined superset of the bits that are set in MOOS bitmap 44 on subsystem 22, while limiting the frequency with which subsystem 22 must instruct subsystem 24 to update the MOOS bitmap.

Fig. 3 is a flow chart that schematically illustrates a method for tracking data storage on system 20, in accordance with an embodiment of the present invention. The method uses bitmaps 40, 42 and 44, as shown in Fig. 2, and is described with reference to these bitmaps. Control unit 30 of subsystem 22 initiates the method whenever host 26 writes data to a specified location on subsystem 22, at a host writing step 70. The location is denoted here as "track E." Control unit 30 places the data in its cache 32, and sets a bit 46 in bitmap 40 to indicate that track E is "dirty" in cache 32, at a data caching step 72. The control unit hardens the data from cache 32 to storage media 34, as noted

above, in a process that take place in background, asynchronously with the host write operation and metadata manipulations that are described here. When subsystem 22 has hardened the data stored in the track corresponding to bit 46, control unit 30 clears the bit in bitmap 40.

After setting bit 46 in bitmap 40, control unit 30 checks bitmap 44 to determine whether the corresponding bit, referred to as MOOS(E), is set in bitmap 44, at a MOOS checking step 74. If MOOS(E) is not set in bitmap 44, control unit 30 updates bitmap 44, at a MOOS update step 76. Typically, when the control unit updates the bitmap, it sets not only MOOS(E) (corresponding to bit 46), but rather a group of bits 50, corresponding to tracks to which host 26 is predicted to direct its subsequent write operations. Any suitable prediction algorithm may be used to select bits 50. For example, bits 50 may comprise E and the next N bits (in the present example, N=3) in bitmap 44 following MOOS(E), as shown in Fig. 2. Control unit 30 may also clear certain bits in bitmap 44 at this stage, as described below with reference to Fig. 4.

After updating bitmap 44, control unit 30 of subsystem 22 transmits a message containing the data that are to be written to track E over link 36 to subsystem 24, at a secondary writing step 78. Alternatively, the message may simply indicate that track E has received write data, and the control unit may send the actual data asynchronously. Upon receiving the data, control unit 30 of subsystem 24 places the data in its local cache 32. The control unit of subsystem 24 checks its own MOOS bitmap to determine whether the bit corresponding to track E is set. If the bit is not set, the control unit

sets a group of bits using the same prediction algorithm as was used in subsystem 22 at step 76. Thus, in the present example, bits 50 will be set in the MOOS bitmap on subsystem 24, as well. Control unit 30 of subsystem 24 then returns a write acknowledgment (ACK), which is received by subsystem 22 at a secondary acknowledgment step 80. Upon receiving this acknowledgment, control unit 30 of subsystem 22 signals its own write acknowledgment to host 26, at a host acknowledgment step 82, and the host write operation is done.

On the other hand, if control unit 30 in subsystem 22 finds at step 74 that MOOS(E) is set, it does not update bitmap 44. Rather, the control unit sets the corresponding bit in bitmap 42, referred to as OOS(E), in an OOS setting step 84. For example, after writing to the track corresponding to bit 46, host 26 may continue writing to the next track, which corresponds to a bit 48 in bitmap 40. Because of the prediction carried out at the previous pass through step 76, the corresponding bit (one of bits 50) is already set in bitmap 44. Thus, the control unit sets OOS(E), and then signals acknowledgment to host 26 immediately at step 82, as described above. In this case, no further operations are required on the MOOS bitmaps at this stage, and the host write operation is completed by subsystem 22 without requiring any synchronous communication with subsystem 24. Synchronous communication is required only when MOOS(E) is not set prior to receiving the host write at step 70.

{In your disclosure, in the middle of page 3, where you talk about how the secondary handles write commands, you say, "if there is a prediction for this data element,

turn it off." I think this point may be incorrect, and I have left it out of the description. As you have described the process, the secondary may turn off the prediction for a data element without the primary knowing that the prediction has been turned off, and without the data necessarily having been hardened on the primary. If the primary fails, this track may never be synchronized. I think that generally speaking, the secondary should clear tracks in its MOOS only when the primary tells it to do so, as described below. Please check.}

Fig. 4 is a flow chart that schematically shows details of MOOS update step 76, in accordance with an embodiment of the present invention. As noted above, when control unit 30 at subsystem 22 determines at step 74 that MOOS(E) is not set, the control unit sets MOOS(E), and also predicts the next tracks to which host 26 is likely to write and sets the corresponding bits in bitmap 44, at a prediction step 90. In the present example, the control unit sets bits MOOS(E) through MOOS(E+N). The number of predicted bits to set, N, is chosen so as to strike the desired balance between low average latency (achieved when N is large) and rapid failure recovery (achieved when N is small, since in this case a relatively smaller number of tracks will be copied back from subsystem 24 to subsystem 22 during recovery). Alternatively, other methods may be used to choose the bits that are to be set in bitmap 44 at step 90. For example, a certain number of bits prior to bit E may be set, in addition to or instead of the bits following E. As noted above, whatever method is used by control unit 30 in subsystem 22, the same method is used by the

control unit in subsystem 24 to update its own MOOS bitmap when it receives data sent from subsystem 22 at step 78.

Referring back to Fig. 2, control unit 30 of subsystem 22 may use generation table 60 to keep track of the bits that have been set (and are subsequently cleared) in bitmap 44. The table shown in the figure comprises generation numbers 62, and for each generation, a bit counter 64 indicating the number of bits in this generation that are set in bitmap 44. When control unit 30 in subsystem 22 sets bits in bitmap 44 at step 90, it also increments counter 64 for the current generation, until the counter reaches a preset maximum. The control unit then goes on to the next generation. Control unit 30 in subsystem 24 uses the same sort of generation counter, with the same maximum counter value, so that each bit that is set in the MOOS bitmaps belongs to the same generation on both subsystems. The control units keep track of the locations of the bits that are set in each generation, by means of further entries (not shown) in table 60, for example, or by a separate listing of bits per generation, or by entering generation values or timestamps in bitmap 44 (in which case "bitmap 44" is no longer simply a bitmap, but rather comprises a more complex metadata record). Counters 64 are decremented as the bits in bitmap 44 are cleared, as described hereinbelow.

As host 26 continues to write data to system 20, more new bits will continue to be set in bitmap 44 on successive iterations through step 76. The greater the number of bits that are set in bitmap 44, while the corresponding tracks on subsystems 22 and 24 are not

actually out of sync, the larger the number of tracks that will be unnecessarily copied from subsystem 24 to subsystem 22 during recovery from failure. In order to limit the number of tracks that are copied unnecessarily, control unit 30 may choose certain tracks to be cleared in bitmap 44, at a bitmap checking step 92 (Fig. 4). The tracks whose bits may be cleared in bitmap 44 are generally those for which the corresponding bits in both of bitmaps 40 and 42 are clear. These tracks are "clean" in cache 32 and are in sync with subsystem 24, meaning that the data stored in these tracks of non-volatile media 34 of subsystems 22 and 24 are substantially identical.

Referring back to Fig. 2, for example, bits 52, 54 and 56 are set in bitmap 44. Bit 56 is also set in bitmaps 40 and 42, and bit 54 is set in bitmap 40. Bits 52, however, are clear in bitmaps 40 and 42, possibly because these bits were set in bitmap 44 at an earlier pass through step 90, based on a prediction (which has gone unrealized) that host 26 would write to these bits. Therefore, the tracks corresponding to bits 52 are known to contain identical data on non-volatile media 34 of subsystems 22 and 24, and these bits may be cleared.

Control unit 30 on subsystem 22 counts the total number, M, of the unnecessarily-set bits in bitmap 44, such as bits 52, and compares this number to a predetermined threshold, at a bitmap evaluation step 94 (Fig. 4). As long as M is below the threshold, there is no need to clear any of the bits in bitmap 44 at this step. The threshold is chosen so as to give the desired balance between low write latency (high threshold) and rapid failure recovery (low threshold). If M is above

the threshold, control unit 30 in subsystem 22 clears some of the unnecessarily-set bits in bitmap 44, at a bit clearing step 96, so that the number of unnecessarily-set bits remaining after this step will be less than the threshold. As noted above, the bits that are cleared in bitmap 44 at this step are selected from among those that are clear in bitmaps 40 and 42, such as bits 52.

Typically, control unit 30 on subsystem 22 keeps a list or other record of the respective times at which the bits in bitmap 42 were set, and chooses at step 94 to clear the unnecessarily-set bits that were set least recently. For example, generation table 60 (Fig. 2) may be used for this purpose. In order to find bits to clear in bitmap 44, the control unit begins with the oldest generation for which counter 64 is non-zero, and determines whether any of the bits that were set in bitmap 44 in this oldest generation are clear in bitmaps 40 and 42. If so, the control unit clears one or more of these bits in bitmap 44. (In the present example, the counter for generation 0, the oldest generation, indicates that four bits are set in bitmap 44 in this generation.) If the control unit does not find a sufficient number of bits to clear in the oldest generation, it move on to the next generation. Upon clearing the bits, the control unit decrements the appropriate generation counter 64 accordingly. Alternatively, other metadata records and/or other criteria may be used to choose the bits to clear at this step.

Upon writing data to subsystem 24 at step 78 after having cleared certain bits in bitmap 44, control unit 30 in subsystem 22 may also instruct the control unit of

subsystem 24 to clear these bits in its own MOOS bitmap. Typically, for efficient maintenance of the MOOS bitmaps and reduction of communication overhead on link 36, control unit 30 of subsystem 22 sends this sort of bit clearing instructions only with respect to entire generations of bits, when the corresponding counter 64 in table 60 has dropped to zero. This technique is described below in greater detail.

Fig. 5 is a flow chart that schematically illustrates a method for maintaining bitmap 44 on subsystem 22 and the maintaining the corresponding MOOS bitmap on subsystem 24, in accordance with an embodiment of the present invention. This method is carried out mainly as a background process, asynchronously with the host write process shown in Fig. 3. It is used by control unit 30 of subsystem 22 in conjunction with the background processes that are employed to harden data from cache 32 to non-volatile media 34 in subsystem 22 and to copy data from subsystem 22 to subsystem 24.

The method of Fig. 5 may be invoked when control unit 30 of subsystem 22 receives a message over link 36 from subsystem 24 acknowledging receipt of data copied over link 36 to a certain track - track E - on subsystem 24, at an acknowledgment step 100. The data copy operation may have been a synchronous write, performed at step 78 (Fig. 3) when the control unit determined that MOOS(E) was clear at step 74. In this case, the corresponding bit in bitmap 42, OOS(E), will be clear when the acknowledgment is received from subsystem 24. In the example shown in Fig. 2 and described above, bit 46 is clear in bitmap 42 for this reason. Therefore, upon receiving the acknowledgment, control unit 30 of

subsystem 22 checks whether OOS(E) is set in bitmap 42, at a first OOS checking step 102. If so, the acknowledgment received at step 100 is the acknowledgment referred to in step 80 in the method of Fig. 3, and the control unit proceeds to acknowledge the host write operation at step 82, without further change to bitmap 44 at this point.

On the other hand, if OOS(E) is found to be set at step 102, control unit 30 of subsystem 22 clears OOS(E) in bitmap 42, at an OOS clearing step 104. The control unit also clears the corresponding bit, MOOS(E), in bitmap 44, at a MOOS clearing step 106. **{As I noted in an e-mail to you, I am not sure this point is correct/complete. It is based on your disclosure, which says: "Whenever the primary gets an acknowledge on writing a data element to the secondary, it turns both is OOS and its prediction off." There may be cases in which the primary receives an acknowledgment from the secondary (and therefore clears OOS(E)) before it has finished hardening the data in the corresponding track. If the primary instructs the secondary to clear the prediction MOOS(E) for this track following the acknowledgment, and then the primary fails before hardening the track, the secondary will not know that this track needs to be copied back to the primary on failback. It seems to me that for safety, you must have both a clean local cache (bitmap 40) and clear OOS(E) before clearing MOOS(E). Please check.}**

The method of Fig. 5 may also be invoked when control unit 30 finishes hardening a track of data on non-volatile media 34 of subsystem 22, at a hardening

step 108. In this case, the control unit also checks whether the corresponding track is still out of sync with subsystem 24, i.e., whether OOS(E) is set in bitmap 42, at a second OOS checking step 110. If OOS(E) is set, control unit 30 on subsystem 22 takes no further action on bitmap 44 at this point. On the other hand, if OOS(E) is found to be clear at step 110, the control unit clears MOOS(E) in bitmap 44 at step 106, as described above, since this track is known to be hardened and synchronized.

Upon clearing a bit in bitmap 44 at step 106, control unit 30 determines the generation to which the bit belongs, and decrements the corresponding generation counter 64 in table 60 (Fig. 2), at a counter decrementation step 112. The control unit checks to determine whether the value of this counter has now dropped to zero, at a counter checking step 114. The counter may drop to zero either as a result of the process shown here in Fig. 5, or as a result of clearing bits in bitmap 44 to reduce the number of unfulfilled predictions at step 96, as described above with reference to Fig. 4. In either case, when counter 64 has dropped to zero, control unit 30 of subsystem 22 sends a message to subsystem 24 identifying the generation that has been cleared in bitmap 44, and instructing subsystem 24 to clear all the bits in the corresponding generation in its own MOOS bitmap, at a generation clearing step 116. As noted above, because both of subsystems 22 and 24 use the same prediction algorithm for setting bits in the MOOS bitmap, and have the same number of predictions in each generation, the same bits will be set in each generation of the MOOS bitmap on both subsystems. The command

instructing subsystem 24 to clear a given generation may be sent as a separate message, or it may be appended to the next data message sent at step 78.

Alternatively, other methods may be used for maintaining synchronization between the MOOS records on subsystems 22 and 24. For example, control unit of subsystem 22 may append to every write message that it sends at step 78 (or at least to some of the messages) an explicit list of bits to be set and cleared in the MOOS bitmap of subsystem 24. Other data structures and messaging methods for this purpose will be apparent to those skilled in the art.

As noted above, the designations "primary" and "secondary" applied to subsystems 22 and 24 are arbitrary. The principles of the present invention can be applied to symmetrical configurations, in which subsystem 22 serves as the primary subsystem for some hosts, while subsystem 24 serves as the primary subsystem for others. In this case, each of subsystems 22 and 24 serves as the secondary subsystem for the other (primary) subsystem, and both subsystems carry out both the primary and secondary sides of the methods described hereinabove. **{But if the secondary fails (in any implementation of the present invention - not only symmetrical configurations), how does the primary know what data it needs to copy to the secondary? The secondary may acknowledge a write to the primary after putting the data in its cache, before it hardens the data. On this basis, the primary will erase the corresponding bit from both OOS and MOOS. If the secondary fails before hardening the data, the primary will not know that this track is out of sync.}**

It will thus be appreciated that the preferred embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and subcombinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.

CLAIMS

1. A method for managing a data storage system that includes primary and secondary storage subsystems, including respective first and second non-volatile storage media, the method comprising:

maintaining a record on the secondary storage subsystem, which is predictive of locations to which data are to be written on the primary storage subsystem by a host processor;

receiving at the primary storage subsystem, from the host processor, the data to be written to a specified location on the first non-volatile storage media;

if the specified location is not included in the record, sending a message from the primary storage subsystem to the secondary storage subsystem so as to cause the secondary storage subsystem to update the record;

signaling the host processor that the data have been stored in the data storage system responsively to receiving the data and, if the specified location was not included in the record, responsively to receiving an acknowledgment at the primary storage subsystem from the secondary storage subsystem indicating that the record has been updated; and

storing the data in the specified location on both the first and second non-volatile storage media.

2. The method according to claim 1, wherein sending the message comprises copying the data synchronously from the primary storage subsystem to the secondary storage subsystem.

3. The method according to claim 2, wherein storing the data comprises, if the specified location is included in the record, copying the data from the primary storage subsystem to the secondary storage subsystem asynchronously, without updating the record with respect to the specified location.

4. The method according to claim 3, wherein copying the data comprises transmitting the data between mutually-remote sites over a communication link between the sites.

5. The method according to claim 4, wherein the second non-volatile storage media are operated by a service provider other than an owner of the primary storage subsystem, and wherein storing the data comprises storing the data on the second non-volatile storage media on a fee-per-service basis.

6. The method according to claim 3, wherein signaling the host processor comprises, if the specified location is included in the record, indicating to the host processor that the data have been stored without waiting to receive the acknowledgment from the secondary storage subsystem.

7. The method according to claim 1, wherein copying the data comprises creating a mirror on the secondary storage subsystem of the data received by the primary storage subsystem.

8. The method according to claim 7, and comprising, upon occurrence of a failure in the primary storage subsystem, configuring the secondary storage subsystem to serve as the primary storage subsystem so as to receive

further data from the host processor to be stored by the data storage system.

9. The method according to claim 7, and comprising, upon recovery of the system from a failure of the primary storage subsystem, conveying, responsively to the record, a portion of the data from the secondary storage subsystem to the primary storage subsystem for storage on the primary storage subsystem.

10. The method according to claim 1, wherein maintaining and updating the record comprise marking respective bits in a bitmap corresponding to the locations to which the data are to be written on the first and second non-volatile storage media.

11. The method according to claim 1, wherein maintaining the record comprises maintaining a first record on the primary storage subsystem and a second record on the secondary storage subsystem, wherein the locations included in the second record are a superset of the locations included in the first record, and wherein sending the message comprises deciding at the primary storage subsystem to send the message responsively to the first record.

12. The method according to claim 11, wherein sending the message comprises modifying both the first and second records responsively to the specified location.

13. The method according to claim 12, wherein modifying both the first and second records comprises adding a plurality of locations, including the specified location, to both the first and second records.

14. The method according to claim 11, wherein maintaining the first and second records comprises removing one or more locations, other than the specified location, from the first record, and instructing the secondary storage subsystem to remove the one or more locations from the second record, so as to limit a size of the second record.

15. The method according to claim 14, wherein storing the data comprises copying the data to be stored in the one or more locations from the primary storage subsystem to the secondary storage subsystem, and wherein removing the one or more locations comprises receiving a return message from the secondary storage subsystem indicating that the secondary storage subsystem has received the copied data, and removing the one or more locations from the record responsively to the return message.

16. The method according to claim 14, wherein removing the one or more locations comprises identifying the locations at which the first and second non-volatile storage media contain substantially identical data, and selecting for removal one of the identified locations that was least-recently added to the first record.

17. The method according to claim 14, wherein sending the message comprises adding one or more entries to both the first and second records responsively to the specified location, and grouping the entries added to the first and second records in generations according to an order of adding the entries to the records, and wherein instructing the secondary storage subsystem to remove the one or more locations comprises determining at the primary subsystem that one of the generations may be

removed from the records, and instructing the secondary storage subsystem to remove the one of the generations from the second record.

18. The method according to claim 14, wherein instructing the secondary storage subsystem to remove the one or more locations comprises appending an instruction to the message sent from the primary storage subsystem to the secondary storage subsystem.

19. The method according to claim 1, wherein sending the message causes the secondary storage subsystem to predict one or more further locations to which the host processor is expected to write the data in a subsequent write operation, and to add the one or more further locations to the record.

20. The method according to claim 19, wherein the one or more further locations comprise a predetermined number of consecutive locations in proximity to the specified location.

21. A data storage system, comprising:

a primary storage subsystem, which comprises first non-volatile storage media; and

a secondary storage subsystem, which comprises second non-volatile storage media, and which is arranged to maintain a record that is predictive of locations to which data are to be written on the primary storage subsystem by a host processor,

wherein the primary storage subsystem is arranged to receive the data from a host processor for writing to a specified location, and to store the data in the specified location on the first non-volatile storage

media while copying the data to the second storage subsystem, which is arranged to store the data in the specified location on the second non-volatile storage media, and

wherein the primary storage subsystem is further arranged, upon receiving from the host processor the data to be written to a specified location on the first non-volatile storage media, if the specified location is not included in the record, to send a message to the secondary storage subsystem so as to cause the secondary storage subsystem to update the record and to return an acknowledgment to the primary storage subsystem indicating that the record has been updated, and

wherein the primary storage subsystem is further arranged to signal the host processor that the data have been stored in the data storage system responsively to receiving the data and, if the specified location was not included in the record, responsively to receiving the acknowledgment from the secondary storage subsystem.

22. The system according to claim 21, wherein the message sent to the secondary storage subsystem comprises the data, which are copied synchronously from the primary storage subsystem to the secondary storage subsystem.

23. The system according to claim 22, wherein the primary storage subsystem is arranged, if the specified location is included in the record, to copy the data from the primary storage subsystem to the secondary storage subsystem asynchronously, without causing the secondary storage subsystem to update the record with respect to the specified location.

24. The system according to claim 23, wherein the first and second non-volatile storage media are located at mutually-remote sites, and wherein at least one of the primary and secondary storage subsystems is arranged to transmit the data over a communication link between the sites.

25. The system according to claim 24, wherein the second non-volatile storage media are operated by a service provider other than an owner of the primary storage subsystem, and are operated on a fee-per-service basis by a service provider other than an owner of the primary storage subsystem.

26. The system according to claim 23, wherein the primary storage subsystem is arranged, if the specified location is included in the record, to signal to the host processor that the data have been stored without waiting to receive the acknowledgment from the secondary storage subsystem.

27. The system according to claim 21, wherein the secondary storage subsystem is arranged to mirror the data held by the primary storage subsystem.

28. The system according to claim 27, wherein upon occurrence of a failure in the primary storage subsystem, the secondary storage subsystem is configurable to serve as the primary storage subsystem so as to receive further data from the host processor to be stored by the data storage system.

29. The system according to claim 27, wherein upon recovery of the system from a failure of the primary storage subsystem, the secondary storage subsystem is

arranged to convey, responsively to the record, a portion of the data from the second non-volatile storage media to the primary storage subsystem for storage on the first non-volatile storage media.

30. The system according to claim 21, wherein the record comprises a bitmap, and wherein the secondary storage subsystem is arranged to mark respective bits in the bitmap corresponding to the locations to which the data are to be written by the host processor.

31. The system according to claim 21, wherein the record maintained by the secondary storage subsystem comprises a second record, and wherein the primary storage subsystem is arranged to maintain a first record indicative of the second record, wherein the locations included in the second record are a superset of the locations included in the first record, and wherein the primary storage subsystem is arranged to determine whether to send the message responsively to the first record.

32. The system according to claim 31, wherein the primary and secondary storage subsystems are arranged to update the first and second records, respectively, responsively to the specified location.

33. The system according to claim 32, wherein the primary and secondary storage subsystems are arranged to update the first and second records by adding a plurality of locations, including the specified location, to both the first and second records.

34. The system according to claim 31, wherein the primary storage subsystem is arranged to remove one or more locations, other than the specified location, from

the first record, and to instruct the secondary storage subsystem to remove the one or more locations from the second record, so as to limit a size of the second record.

35. The system according to claim 34, wherein the secondary storage subsystem is arranged to send a return message to the primary storage subsystem, indicating that the secondary storage subsystem has received the copied data, and wherein the primary storage subsystem is arranged to remove the one or more locations from the record responsively to receiving the return message.

36. The system according to claim 34, wherein the primary storage subsystem is arranged to identify the locations at which the first and second non-volatile storage media contain substantially identical data, and to select for removal one of the identified locations that was least-recently added to the first record.

37. The system according to claim 34, wherein the primary and secondary storage subsystems are arranged to respectively add one or more entries to both the first and second records responsively to the specified location, and to group the entries added to the first and second records in generations according to an order of adding the entries to the records, and wherein the primary storage subsystem is arranged to determine that one of the generations may be removed from the records, and to instruct the secondary storage subsystem to remove the one of the generations from the second record.

38. The system according to claim 34, wherein the primary storage subsystem is arranged to append an

instruction to the message sent to the secondary storage subsystem, so as to instruct the secondary storage subsystem to remove the one or more locations from the second record.

39. The system according to claim 21, wherein the message causes the secondary storage subsystem to predict one or more further locations to which the host processor is expected to write the data in a subsequent write operation, and to add the one or more further locations to the record.

40. The system according to claim 39, wherein the one or more further locations comprise a predetermined number of consecutive locations in proximity to the specified location.

41. A computer software product for use in a data storage system including primary and secondary storage subsystems, which include respective first and second control units and respective first and second non-volatile storage media, the product comprising a computer-readable medium in which program instructions are stored, which instructions, when read by the first and second control units, cause the first control unit to receive data from a host processor for writing to a specified location, and to store the data in the specified location on the first non-volatile storage media while copying the data to the second storage subsystem, and cause the second control unit to maintain a record that is predictive of locations to which the data are to be written on the primary storage subsystem by the host processor, and to store the data copied to

the second storage subsystem in the specified location on the second non-volatile storage media,

wherein the instructions further cause the first control unit, if the specified location is not included in the record, to send a message to the secondary storage subsystem so as to cause the second control unit to update the record and to return an acknowledgment to the primary storage subsystem, and cause the first control unit to signal the host processor that the data have been stored in the data storage product responsively to receiving the data and, if the specified location was not included in the record, responsively to receiving the acknowledgment from the second control unit.

42. The product according to claim 41, wherein the message sent to the secondary storage subsystem comprises the data, which are copied synchronously from the primary storage subsystem to the secondary storage subsystem.

43. The product according to claim 42, wherein the instructions cause the first control unit, if the specified location is included in the record, to copy the data from the primary storage subsystem to the secondary storage subsystem asynchronously, without causing the second control unit to update the record with respect to the specified location.

44. The product according to claim 43, wherein the first and second non-volatile storage media are located at mutually-remote sites, and wherein the instructions cause at least one of the first and second control units to transmit the data over a communication link between the sites.

45. The product according to claim 44, wherein the second non-volatile storage media are operated by a service provider other than an owner of the primary storage subsystem, and are operated on a fee-per-service basis by a service provider other than an owner of the primary storage subsystem.

46. The product according to claim 43, wherein the instructions cause the first control unit, if the specified location is included in the record, to signal to the host processor that the data have been stored without waiting to receive the acknowledgment from the second control unit.

47. The product according to claim 41, wherein the instructions cause the first and second control units to mirror the data held by the primary storage subsystem on the secondary storage subsystem.

48. The product according to claim 47, wherein the instructions cause the secondary storage subsystem, upon occurrence of a failure in the primary storage subsystem, to serve as the primary storage subsystem so as to receive further data from the host processor to be stored by the data storage system.

49. The product according to claim 47, wherein upon recovery of the system from a failure of the primary storage subsystem, the instructions cause the second control unit to convey, responsively to the record, a portion of the data from the second non-volatile storage media to the primary storage subsystem for storage on the first non-volatile storage media.

50. The product according to claim 41, wherein the record comprises a bitmap, and wherein the instructions cause the second control unit to mark respective bits in the bitmap corresponding to the locations to which the data are to be written by the host processor.

51. The product according to claim 41, wherein the record maintained by the second control unit comprises a second record, and wherein the instructions cause the first control unit to maintain a first record indicative of the second record, wherein the locations included in the second record are a superset of the locations included in the first record, and wherein the instructions cause the first control unit to determine whether to send the message responsively to the first record.

52. The product according to claim 51, wherein the instructions cause the first and second control units to update the first and second records, respectively, responsively to the specified location.

53. The product according to claim 52, wherein the instructions cause the first and second control units to update the first and second records by adding a plurality of locations, including the specified location, to both the first and second records.

54. The product according to claim 51, wherein the instructions cause the first control unit to remove one or more locations, other than the specified location, from the first record, and to instruct the second control unit to remove the one or more locations from the second record, so as to limit a size of the second record.

55. The product according to claim 54, wherein the instructions cause the second control unit to send a return message to the primary storage subsystem, indicating that the secondary storage subsystem has received the copied data, and wherein the instructions cause the first control unit to remove the one or more locations from the record responsively to receiving the return message.

56. The product according to claim 54, wherein the instructions cause the first control unit to identify the locations at which the first and second non-volatile storage media contain substantially identical data, and to select for removal one of the identified locations that was least-recently added to the first record.

57. The product according to claim 54, wherein the instructions cause the first and second control units to respectively add one or more entries to both the first and second records responsively to the specified location, and to group the entries added to the first and second records in generations according to an order of adding the entries to the records, and wherein the instructions cause the first control unit to determine that one of the generations may be removed from the records, and to instruct the second control unit to remove the one of the generations from the second record.

58. The product according to claim 54, wherein the instructions cause the first control unit to append an instruction to the message sent to the secondary storage subsystem, so as to instruct the second control unit to remove the one or more locations from the second record.

59. The product according to claim 41, wherein the instructions cause the second control unit, responsively to the message, to predict one or more further locations to which the host processor is expected to write the data in a subsequent write operation, and to add the one or more further locations to the record.

60. The product according to claim 59, wherein the one or more further locations comprise a predetermined number of consecutive locations in proximity to the specified location.

STORAGE DISASTER RECOVERY USING A PREDICTED SUPERSET OF UNHARDENED PRIMARY DATA

ABSTRACT

FIG. 1

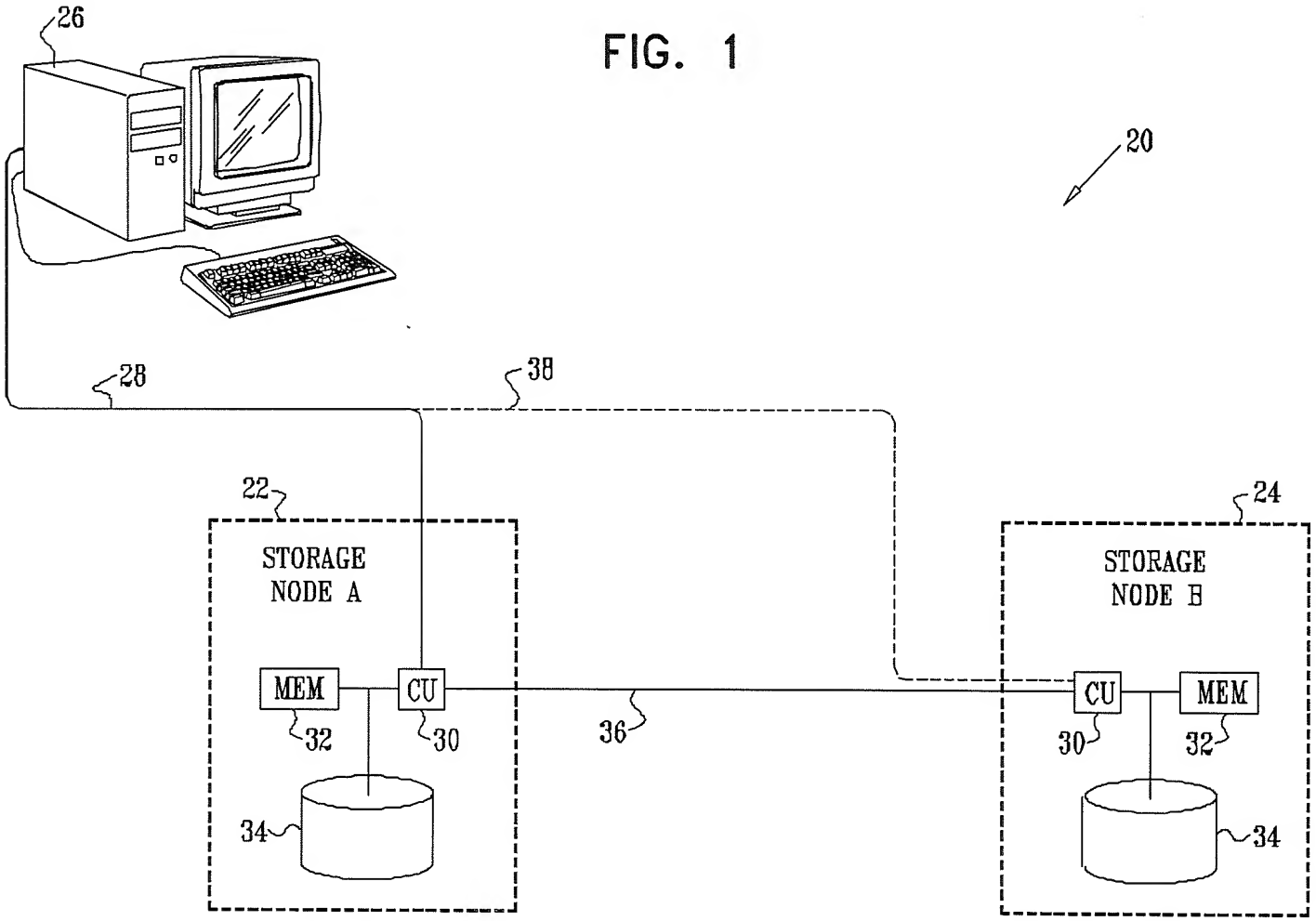
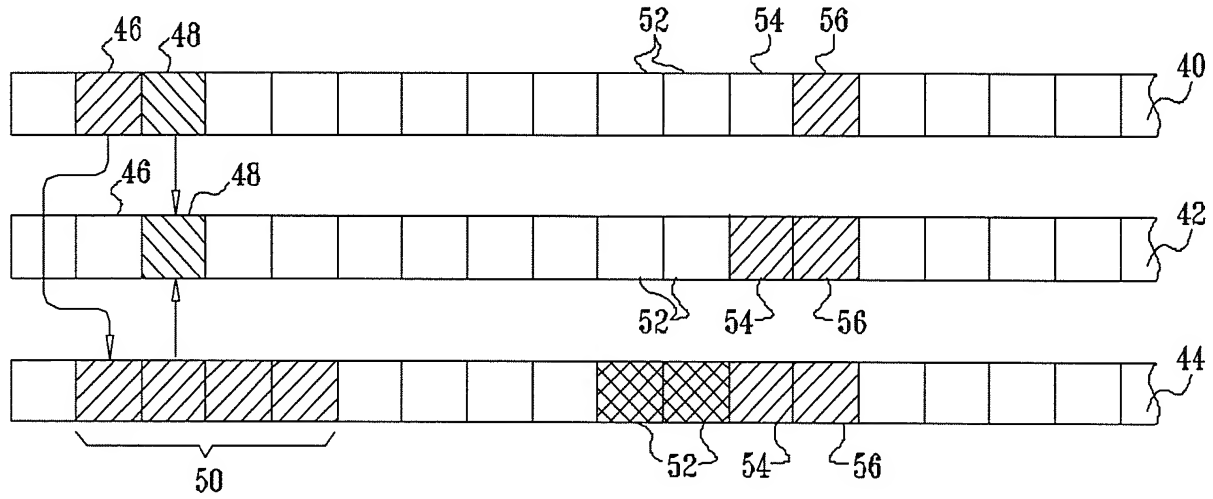

26
20
28
38
22
24
STORAGE
NODE A
STORAGE
NODE B
MEM
CU
CU
MEM
32
30
36
30
32
34
34

FIG. 2



GEN	#MOOS
0	4
1	7
2	12
3	0

60
62
64

FIG. 3

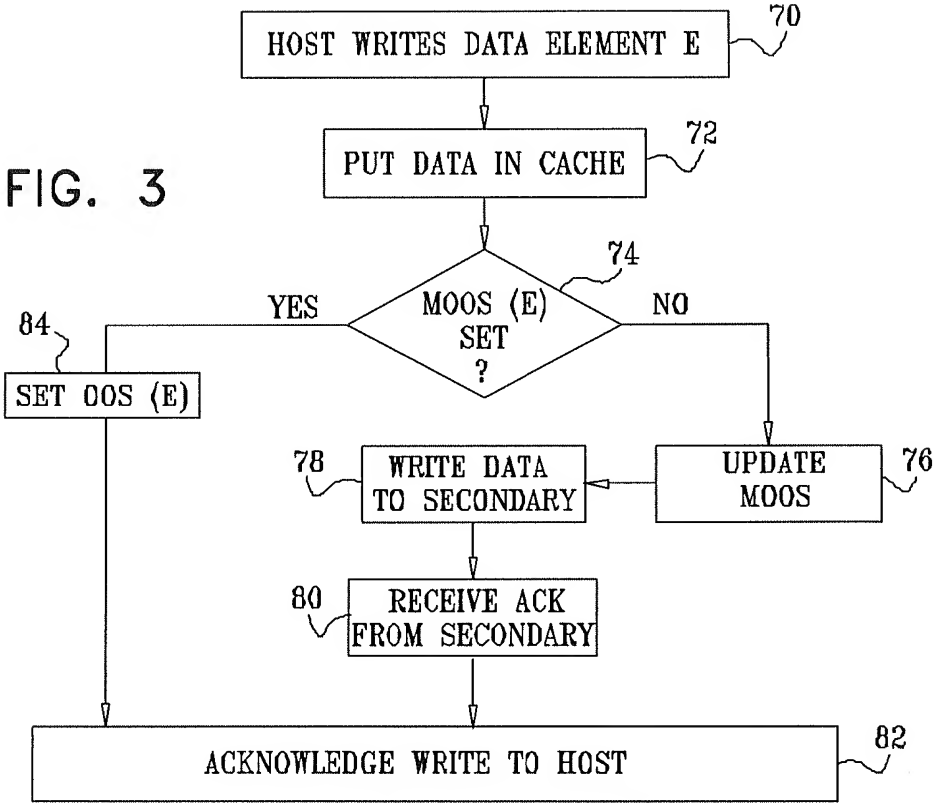

HOST WRITES DATA ELEMENT E
70
PUT DATA IN CACHE
72
MOOS (E) SET ?
74
YES
NO
84
SET OOS (E)
78
WRITE DATA TO SECONDARY
UPDATE MOOS
76
80
RECEIVE ACK FROM SECONDARY
ACKNOWLEDGE WRITE TO HOST
82

FIG. 4

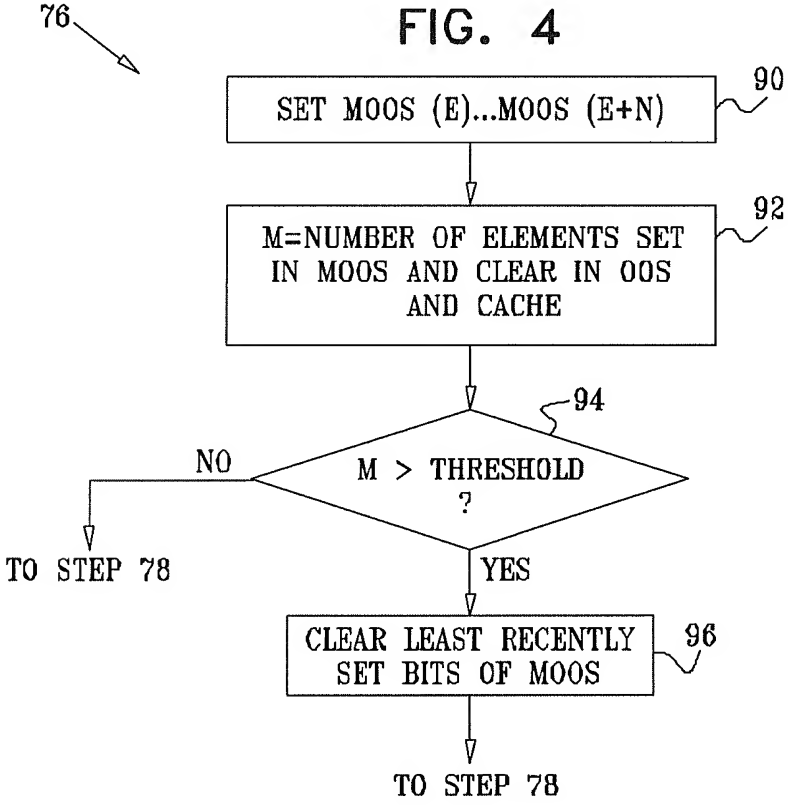

76
SET MOOS (E)...MOOS (E+N)
90
M=NUMBER OF ELEMENTS SET IN MOOS AND CLEAR IN OOS AND CACHE
92
94
M > THRESHOLD ?
NO
TO STEP 78
YES
CLEAR LEAST RECENTLY SET BITS OF MOOS
96
TO STEP 78

FIG. 5

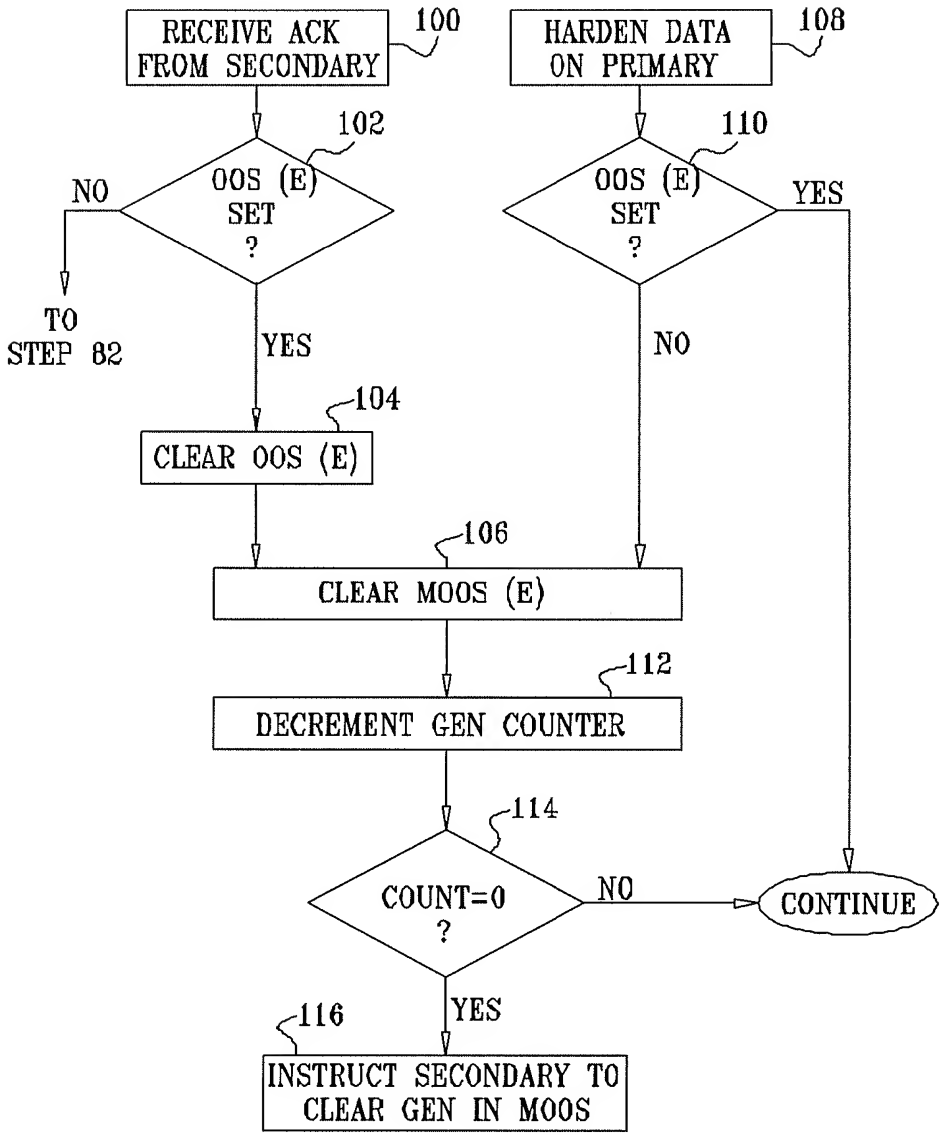

RECEIVE ACK FROM SECONDARY
100
HARDEN DATA ON PRIMARY
108
102
OOS (E) SET ?
NO
TO STEP 82
YES
110
OOS (E) SET ?
YES
NO
104
CLEAR OOS (E)
106
CLEAR MOOS (E)
112
DECREMENT GEN COUNTER
114
COUNT=0 ?
NO
CONTINUE
YES
116
INSTRUCT SECONDARY TO CLEAR GEN IN MOOS

Exhibit B

From: Daniel Kligler
To: Suzanne Erez
Date: 9/4/03 6:44AM
Subject: Re: IL9-2003-0031, your 49265 - IBM CONFIDENTIAL

Dear Suzanne,

Attached please find a final, clean draft of this application. I have incorporated Aviad's correction as noted below, but I have not changed the term "selecting." I do not remember what trouble we may have had with an examiner in this regard, although it sounds as though it may have been a 101 rejection. In the present case, the claims say explicitly "selecting for removal," and this step is followed by a step of actually removing, so that I do not think we will have a problem or that we will gain anything by "marking."

In preparation for filing, we have also added an abstract and paraphrased the claims in the Summary of the Invention.

As I noted earlier, this application should be filed together with -0032. Before filing this application, you should fill in the filing date of -0033 on page 15.

I will send you the printed figures for these three applications (-0031, 32 and 33) by mail. Please let me know if you need us to do any further work on these cases, or whether we can now close our files.

Regards,
Daniel

>>> "Aviad Zlotnick" <AVIAD@il.ibm.com> 09/03/03 02:52PM >>>

Danny,

Here are my remarks:

Page 23, line 7, from 2nd word, reads "it move on to". "move" s/b "moves"

I believe we had a problem with an examiner that thought that there is not enough substance in "selecting". Should we use "marking for later removal"?

Apart from this, I think the application can be filed.

Thank you again,

Aviad

Tel: +972-48-296-284, Cell: +972-66-976-284, Fax: +972-48-296-112, Email: aviad@il.ibm.com

CC: HRL_IPDEPARTMENT@il.ibm.com; Zlotnick, Aviad

Exhibit C

From: Ronen Harel
To: Suzanne Erez
Date: 9/15/03 9:40AM
Subject: Ref: 49264, 49265, 49266,49267

Good Morning Suzanne & Daniel
Drawings For Files 4926-4,5,6,7
Ronen

>>> "Suzanne Erez" <SUZANNE@il.ibm.com> 09/14/03 03:38PM >>>

Daniel - thank you for catching that, you are correct. Please add US1.

Suzanne Erez
IP Department
IBM Haifa Laboratories
Tel: 972-4-829-6069 Fax: 972-4-829-6521
suzanne@il.ibm.com

If you think that you can think about a thing, inextricably attached to something else,
without thinking of the thing it is attached to, then you have a legal mind.
- Thomas Reed Powell

"Daniel Kligler" <dkligler@stc.co.il> To: Suzanne Erez/Haifa/IBM@IBMIL
cc: HRL IP Department/Haifa/IBM@IBMIL, "Ronen Harel" <RonenH@stc.co.il>
Subject: Re: Fw: IL920030031US1, (49265) 32US1 (49266) and 33US1 (49267)
14-09-03 04:28 PM

Dear Suzanne,

Does this also mean that you want us to add "US1" at the end of every docket number in the figure header?

Regards,
Daniel

>>> "Suzanne Erez" <SUZANNE@il.ibm.com> 09/14/03 08:34AM >>>

(See attached file: heading.doc)
Hello Daniel,
Pleas see the letter below. IBM Watson could not file the patent applications due to the figures. Please add to the header of each figure the initials SCK, as in the document above. The initials did appeared on the sample sent to Ronen, and perhaps he did not understand the significance.
Thanks
Suzanne

Suzanne Erez
IP Department
IBM Haifa Laboratories
Tel: 972-4-829-6069 Fax: 972-4-829-6521
suzanne@il.ibm.com

Any sufficiently advanced technology is indistinguishable from magic.
- Murphy's Technology Laws
----- Forwarded by Suzanne Erez/Haifa/IBM on 14-09-03 09:25 AM -----